m-Stability: Threshold Security Meets Transferable Utility

Osman Biçer
Koç University
obicer17@ku.edu.tr

Burcu Yıldız
Koç University
byildiz17@ku.edu.tr

Alptekin Küpçü
Koç University
akupcu@ku.edu.tr

September 29, 2021

Abstract

Use of game theory and mechanism design in cloud security is a well-studied topic. When applicable, it has the advantages of being efficient and simple compared to cryptography alone. Most analyses consider two-party settings, or multi-party settings where coalitions are not allowed. However, many cloud security problems that we face are in the multi-party setting and the involved parties can almost freely collaborate with each other. To formalize the study of disincentivizing coalitions from deviating strategies, a well-known definition named k-resiliency has been proposed by Abraham et al. (ACM PODC '06). Since its proposal, k-resiliency and related definitions are used extensively for mechanism design. However, in this work we observe the shortcoming of k-resiliency. That is, although it is secure, these definitions are too strict to use for many cases and rule out secure mechanisms as insecure. To overcome this, we propose a new definition named ℓ-repellence against the presence of a single coalition to replace k-resiliency. Our definition incorporates transferable utility in game theory as it is realistic in many distributed and multi-party computing settings. We also propose m-stability definition against the presence of multiple coalitions, which is inspired by threshold security in cryptography. We then show the advantages of our novel definitions on three mechanisms, none of which were previously analyzed against coalitions: incentivized cloud computation, forwarding data packages in ad hoc networks, and connectivity in ad hoc networks. Regarding the former, our concepts improve the proposal by Küpçü (IEEE TDSC '17), by ensuring a coalition-proof mechanism.

Keywords: threshold security, cooperative game theory, outsourced computing, ad hoc networks

1 Introduction

Game theory has been applied to many areas including social sciences, economics, biology, law, and cloud security. Regarding the latter, application of game theory has been shown rather useful. This is because it usually leads to more efficient solutions for problems for which impractical protocols have been proposed [1].

Unlike cryptographic protocols, game theory based solutions depend more on the rationality of the involved parties [2]. It can be seen that in many computer and network security situations, one may assume that malicious actions will not be taken, if they harm the one who takes them. Therefore, one may reduce the threats that she will countermeasure against to those that are

beneficial to the attacker. This is especially true for cloud security, as cloud companies usually care about their reputation, which directly affects their income, and hence are likely to act rationally rather than directly maliciously.

Game theory has a large application area in security, including incentivized outsourced cloud computing [3, 4, 5, 6], distributed systems and consensus protocols [7, 8, 9, 10], network security and routing [11, 12], vehicular networks [13], distributed file sharing [14]. Most of these works either consider simple two-party settings, or multi-party settings where coalitions are not allowed. However, especially in online settings, coalitions may be a more prominent issue, as it is hard to disincentivize each party by reputation loss, and sometimes the use of public mechanisms such as the blockchain directly enables coalitions (knowingly or unknowingly) [6]. As formalization of the studies against coalitions with up to k parties, k-resiliency definition [7] has been influential. The authors also proposed its extension as (k,t)-robustness to cover malicious players. These definitions have been used by several works, including [15, 8, 16, 17]. However, a closer look on these definitions reveals that they have some shortcomings: (i) they are too strict for practice, and mark some secure mechanisms as insecure, (ii) they are not designed against the presence of multiple coalitions. These observations led us to further investigate k-resiliency and (k,t)-robustness definitions to provide improved and practical definitions taking into account potential multiple coalitions' coexistence in the system. We list the main contributions of this paper as follows:

- To resolve the issue (i) with k-resiliency and (k,t)-robustness definitions, we propose ℓ-repellence against the presence of a single coalition (replacing k-resiliency). If additionally there are arbitrarily deviating players, our proposed (ℓ,t)-resistance replaces (k,t)-robustness. Our definitions incorporate transferable utility assumption in game theory, as it is realistic in many distributed and multi-party computing settings.

- To cope with the issue (ii), we propose m-stability definition against the presence of multiple coalitions. Inspired by threshold security in cryptography [18, 19], our m-stability definition protects the system against deviation of any number of rational coalitions as long as none of them has more than m members. Note that this is novel, as previous definitions considered only a single deviating coalition existing in the system.

- On three already existing mechanisms, one for incentivized cloud computing, one for forwarding data packages in ad hoc networks, and one for connectivity in ad hoc networks, we separately show applicability and advantage of our novel definitions. We note that none of these works has been analyzed against coalitions previously.

- Regarding incentivized cloud computing, our definitions improve the multi-party mechanism of [4] by fine tuning parameter selection to achieve security against large coalitions of size up to one less than all the contractors involved. More concretely, we set tighter relative bounds for rewards and bounties given to the contractors.

2 Related Work

Mechanism Design. The use of mechanism design in distributed systems and multi-party computation is an extensively studied subject with some notable works including [20, 21, 7, 3, 22, 23, 8, 24, 4]. There are some proposed models such as the Byzantine (B), altruistic (A),

rational (R) or BAR model [25, 26], type model [27], and the rational protocol design (RPD) framework [28]. In this work, for simplicity, we model the parties as BAR model (without altruistic players) rather than the type model. We do not utilize RPD framework, as we are interested in internal deviations of the participants, instead of an outlier adversary taking control of the protocol participants.

Security against Coalitions. For coalition-proofness in the realm of distributed systems and multi-party computation, conventionally k-resiliency and its extended version (k,t)-robustness in presence of Byzantine parties [7] are used. A line of works [15, 8, 16, 17] confronts to satisfy this definition. There also exist models as in [26, 29] based on simpler coalitional abilities such as "cheap talk" [30]. In this work, we propose ℓ-repellence, (ℓ,t)-resistance, and m-stability definitions as more flexible proof definitions, instead of the conventional k-resiliency and the related (k,t)-robustness [7] definitions.

Cryptocurrencies and Smart Contracts. Thanks to the inspiration by Bitcoin [31], many cryptocurrencies (e.g., Ethereum [32], Litecoin [33]) based on blockchain have been proposed and gone alive. There exist various consensus methods that are used for blockchain security (e.g., proof of work [34], proof of stake [35], proof of validation [36], or Byzantine fault tolerance [37, 38]). Moreover, there exist other decentralization techniques in cryptocurrencies, such as, IOTA [39] using "The Tangle" where the data is kept as a directed acyclic graph, Phantom [40] and Spectre [41] where the data is kept as a directed acyclic graph of blocks "blockDAG".

A smart contract is utilized for establishing binding contracts in cryptocurrency environments. It is publicly executed by the underlying consensus protocol. First, it collects the submitted transactions to the network and then appends them to the blockchain. We assume SC as a trusted third party (TTP) just as in other smart contract based protocols [10, 42, 5, 6]. We note that in Ethereum any Turing machine can be run as smart contract as long as the generator pays the price per instruction as "gas" [43]. However, in Bitcoin, the programming language for SC generation is not Turing complete, and not all computer programs can be run [44]. We refer to [43, 44, 45] for further investigation of smart contracts.

3 Background on Game Theory and Mechanism Design

An n-player game can be defined by a set of players $N = (P_1, \ldots, P_n)$, the m_i possible actions $\mathcal{A}_i = (A_i^1, \ldots, A_i^{m_i})$ of each player $i \in N$ and a utility function U_i of each player $i \in N$ which assigns a real-value for each possible action vector A that specifies one action for every player in the game. The goal of each player is to maximize her utility. A strategy s_i of player i is a probability distribution over the possible actions of the player i. We denote by $s_i(a)$ the probability the strategy s_i assigns to the action a. Note that mechanisms are designed to incentivize each player to play a particular strategy. In these mechanisms, a specific action can also be referred to as a strategy. Moreover, we denote by $s_D = (s_{P_a}, \ldots, s_{P_b})$ a strategy profile (an ordered set) comprising the strategies of all the players in the set $D = (P_a, \ldots, P_b)$. We highlight that in many games, some of the strategies that can be chosen by separate players are named the same for simplicity, but as their players are different, they are different and still can be combined in a set as separate elements. Further, we denote by $s_{-i} = s - (s_i)$ the strategy profile of all the players excluding the player i. Similarly, s_{-D} denotes the strategy profile of all the players in a game that are not in D. Given a strategy profile s_N, we denote by $U_i(s_N)$ the

expected utility of the player i, if the players in the game play the strategy profile s_N.

Nash Equilibrium. Game theory provides various tools for predicting the outcomes of a game, among which the most commonly used one is the Nash equilibrium which is defined as follows:

Definition 1 (Nash Equilibrium)**.** *For any game with the set N of players, a strategy set $s_N = (s_{P_1}, \ldots, s_{P_n})$ is a Nash equilibrium if $\forall i \in N\ \forall s'_i \neq s_i\ U_i(s_N) \geq U_i(s'_i \cup s_{-i})$.*

Weakly Dominant Strategy. Another useful notion in the analysis of the games is weakly dominant strategy, which is defined for a player as follows:

Definition 2 (Weakly Dominant Strategy of a Player)**.** *A strategy s_i of a player i is its weakly dominant strategy if for all strategies $s'_i \neq s_i$ of i and for all strategy profiles s'_{-i} of players other than i, $U_i(s_i \cup s'_{-i}) \geq U_i(s'_i \cup s'_{-i})$.*

Unlike a Nash equilibrium, a weakly dominant strategy may not always exist in a game. Yet, a strategy set $s_N = (s_{P_1}, \ldots, s_{P_n})$ where each s_i is the weakly dominant strategy of i is a Nash equilibrium, in which case it can be referred to as weakly dominant strategy equilibrium. The definition of weakly dominant strategy of a coalition follows the definition of coalition utility below.

Transferable Utility. In many games involving cooperations, it would be safe to assume that players can engage in binding agreements for how to share the outcome of a game. In particular, if there exists an available currency to the participants, and this currency is valued equally among them, then this assumption becomes more realistic. We stress that due to the increasing use of cryptocurrencies along with smart contracts, nowadays this assumption is realistic in many problems arising in the realm of computer science. The conventional name given to this assumption is transferable utility assumption [46]. This simplifies the analysis by permitting to define a single utility for a coalition as a whole, and by abstracting out how this utility is shared among the participants internally. The utility of a coalition is defined as follows:

Definition 3 (Coalition Utility)**.** *Let s be a strategy set for a game Π played by the players in a mechanism. Then, the utility $U_C(s)$ of a coalition C is defined as $U_C(s) = \sum_{i \in C} U_i(s)$.*

Weakly dominant strategy of a coalition is defined similar to that of a player as below:

Definition 4 (Weakly Dominant Strategy of a Coalition)**.** *A strategy profile s_C of a coalition C is its weakly dominant strategy if for all strategy profiles $s'_C \neq s_C$ of C and for all strategy profiles s'_{-C} of the players outside C, the following holds: $U_C(s_C \cup s'_{-C}) \geq U_i(s'_C \cup s'_{-C})$.*

We stress that the transferable utility assumption is also useful in cases where a rational adversarial party joins a protocol with multiple identities, e.g., as in Sybil attack in peer-to-peer systems [47].

Mechanism Design. Mechanism Design (ME) can be considered as an application of game theory to achieve a goal by incentivizing the rational players for a particular strategy set s_N, where N is the set of all players. Given some desired functionality $\mathcal{F}$, we say that (Π, s_N) is a mechanism for $\mathcal{F}$ if the outcome of (Π, s_N) satisfies $\mathcal{F}$ and the players are incentivized to play s_N.

BAR Model. First, proposed by [25], Byzantine (B), altruistic (A), and rational (R) model (or BAR model) is the commonly used model in mechanism design for distributed systems. The

model defines three types of players: the Byzantine ones (i.e., the ones that can choose any possible strategy, no matter what their utilities are), the altruistic ones (i.e., the ones that always choose the honest strategy, no matter what their utilities are), and the rational ones (i.e., the ones that always choose the strategy that maximizes their utilities). The model itself does not provide a distribution for these types of players in a given game, but rather is useful for asserting and proving statements such as a desirable functionality can be achieved as long as the number of rational and or Byzantine players are below certain bounds.

k-Resiliency. [7] proposes the k-resilient, t-immune, and (k,t)-robust mechanism definitions to protect the mechanism outcome against coalitions and Byzantine players as follows:

Definition 5 (k-Resilient Mechanism)**.** *Given a player set $C \subseteq N$, s_C is a group best response for C to s_{-C}, if for all strategies s'_C played by C and $\forall i \in C$, we have $U_i(s_C \cup s_{-C}) \geq U_i(s'_C \cup s_{-C})$. A joint strategy s_N is a k-resilient equilibrium, if $\forall C \subseteq N$ with $|C| \leq k$, s_C is a group best response for C to s_{-C}, where $s_N = s_C \cup s_{-C}$. Given some desired functionality $\mathcal{F}$, we say that (Π, s_N) is a k-resilient mechanism for $\mathcal{F}$, if s_N is a k-resilient equilibrium of Π and the outcome of (Π, s_N) satisfies $\mathcal{F}$.*

Definition 6 (t-Immune Mechanism)**.** *A joint strategy s_N is a t-immune equilibrium, if $\forall T \subseteq N$ with $|T| \leq t$, for all strategies s'_T played by the players in T, and $\forall i \notin T$, we have $U_i(s'_T \cup s_{-T}) \geq U_i(s_N)$, where $s_N = s_T \cup s_{-T}$. Given some desired functionality $\mathcal{F}$, we say that (Π, s_N) is a t-immune mechanism for $\mathcal{F}$, if s_N is a t-resilient equilibrium of Π and the outcome of (Π, s_N) satisfies $\mathcal{F}$.*

Definition 7 ((k,t)-Robust Mechanism)**.** *A joint strategy s_N is a (k,t)-robust equilibrium, if $\forall C, T \subseteq N$ s.t. $C \cap T = \emptyset$, $|C| \leq k$, and $|T| \leq t$, for all strategies s'_T played by the players in T, and for all strategies s'_C played by C, and $\forall i \in C$, we have $U_i(s_{-T} \cup s'_T) \geq U_i(s_{-(C \cup T)} \cup s'_C \cup s'_T)$, where $s_N = s_C \cup s_T \cup s_{-(C \cup T)}$. Given some desired functionality $\mathcal{F}$, we say that (Π, s_N) is a (k,t)-robust mechanism for $\mathcal{F}$, if s_N is a (k,t)-robust equilibrium of Π and the outcome of (Π, s_N) satisfies $\mathcal{F}$.*

4 Shortcomings of Existing Definitions

In this section, we show some limitations of the existing k-resiliency and (k,t)-robustness definitions for security against coalitions. That is, we provide some example hypothetical games where these definitions indeed fail to satisfy expectations by making it too hard to satisfy (i.e., resulting unnecessary hardness in practice). Also, we elaborate on the limitations of some well-known notions from cooperative game theory for use in security, i.e., they fail to capture coalition strategies well enough (i.e., resulting in security breaches). None of the games provided in this section has been deduced from any specific known game, instead, we have composed them to clarify our argument. Yet, due to the simplicity of these games, they are likely to appear in mechanisms from real life or literature. We show the drawbacks of k-resiliency with existing mechanisms from computer science literature in Section 6.

k-Resiliency is too Strict. Suppose that one needs a mechanism played by a player set N with a desired strategy set s_N. The mechanism has already shown to be Nash equilibrium. Yet, it is needed to be stable against potential coalitions of any 2 players. W.l.o.g., we are interested in the coalition strategies of two particular players, named Alice and Bob. Assuming all the other players play the honest strategy $s_{-(A \cup B)}$, Table 1 shows the utilities (u_A, u_B) of Alice and

Bob from honest strategies s_A and s_B, and deviant strategies s'_A and s'_B, respectively. Notice that the weakly dominant strategies of Alice and Bob are (s_A, s_B).

Alice \ Bob	s_B	s'_B
s_A	$(5,5)$	$(7,2)$
s'_A	$(2,7)$	$(4,4)$

Table (1) The utilities (u_A, u_B) of Alice and Bob from honest strategies s_A and s_B, and deviant strategies s'_A and s'_B.

According to the k-resiliency definition given in Definition 5, this mechanism is not even 2-resilient as there exists a coalition $C =$ (Alice,Bob), with at least one deviant coalition strategy s'_C such that at least one member of the coalition has a greater utility than the one obtained from (s_A, s_B). In fact, there exist two such strategies (s_A, s'_B) and (s'_A, s_B). That is, compared to the honest strategy, the former delivers greater utility for Alice, and the latter is more beneficial to Bob. However, assuming both Alice and Bob are in the coalition for rational purposes, neither Alice would play s_A, nor Bob would play s_B, as otherwise their utilities would decrease. Therefore, there exist games where k-resiliency is too strict to achieve, yet still secure against a coalition based on individual rationality assumption.

The Definitions from Cooperative Game Theory are too Gentle. The cooperative game theory and mechanism design overcomes this issue by notions such as "strong Nash equilibrium" [48] and "coalition-proof Nash equilibrium" [49]. These notions are based on Pareto-optimality, i.e., for their satisfaction there should not be any coalition C with a strategy s'_C that delivers at least the same utility as s_C to each player in C and greater utility than s_C to at least one player in C. We could corporate a definition based on these, but we identify the following issue. Let us change the utility matrix of Alice and Bob in the above mentioned game as in Table 2, assuming all the other parameters are kept unchanged. Notice that the weakly dominant strategies of Alice and Bob are still (s_A, s_B).

Alice $\diagdown$ Bob	s_B	s'_B
s_A	$(5,5)$	$(\mathbf{10},2)$
s'_A	$(2,7)$	$(4,4)$

Table (2) The utilities (u_A, u_B) of Alice and Bob from honest strategies s_A and s_B, and deviant strategies s'_A and s'_B.

The problem with this mechanism is that Alice can offer his coalition partner Bob a transfer of value 4 from his account to hers, if he plays s'_B. Then, she would play s_A and their utility would become $(6,6)$, which beats the utility $(5,5)$ from the honest strategy. Unfortunately, due to the following reasons, this example issue is significant for mechanism design for multi-party protocols and distributed systems. First, with advent and prevalence of cryptocurrencies, the smart contract schemes that can enforce such binding agreements are now available to anyone who can connect to internet. Second, many of the arising problems involve utilities strictly built upon costs, fines, rewards, etc., which can be easily converted to monetary utilities.

This issue is named transferable utility and handled with the notion "core property" [50] in cooperative game theory. Essentially, a strategy set has core property if no coalition can have a greater total utility from another strategy. Our definitions combine this idea with threshold security [18, 19] idea as in k-resiliency, and improve upon it by including Byzantine parties that

can arbitrarily deviate.

5 Our Security Definitions Against Coalitions

In this section, we propose our definitions to replace k-resiliency and (k,t)-robustness, due to the shortcomings of them mentioned in Section 4.

5.1 Security Definitions Against a Single Coalition

First we provide ℓ-repellent mechanism definition, as a direct replacement of k-resiliency.

Definition 8 (ℓ-Repellent Mechanism)**.** *Given a player set $C \subseteq N$, s_C is a best collective response for C to s_{-C}, if for all strategies s'_C played by C, we have $U_C(s_C \cup s_{-C}) \geq U_C(s'_C \cup s_{-C})$. A joint strategy s_N is an ℓ-repellent equilibrium, if $\forall C \subseteq N$ with $|C| \leq \ell$, s_C is a best collective response for C to s_{-C}, where $s_N = s_C \cup s_{-C}$. Given some desired functionality $\mathcal{F}$, we say that (Π, s_N) is an ℓ-repellent mechanism for $\mathcal{F}$, if s_N is an ℓ-repellent equilibrium of Π and the outcome of (Π, s_N) satisfies $\mathcal{F}$.*

In line with the previous (k,t)-robustness definition, we also provide (ℓ,t)-resistant mechanism definition to comprise cases where a collaboration of a set of players and a set of arbitrarily acting players coexist. The following definition is expected to be used in mechanism design instead of (k,t)-robustness.

Definition 9 ((ℓ,t)-Resistant Mechanism)**.** *A joint strategy s_N is an (ℓ,t)-resistant equilibrium, if $\forall C, T \subseteq N$ s.t. $C \cap T = \emptyset$, $|C| \leq \ell$, and $|T| \leq t$, for all strategies s'_T played by the players in T, and for all strategies s'_C played by C, we have $U_C(s_{-T} \cup s'_T) \geq U_C(s_{-(C \cup T)} \cup s'_C \cup s'_T)$, where $s_N = s_C \cup s_T \cup s_{-(C \cup T)}$. Given some desired functionality $\mathcal{F}$, we say that (Π, s_N) is an (ℓ,t)-resistant mechanism for $\mathcal{F}$, if s_N is an (ℓ,t)-resistant equilibrium of Π and the outcome of (Π, s_N) satisfies $\mathcal{F}$.*

5.2 Extension Against Multiple Coalitions

We extend our ℓ-repellence and (ℓ,t)-resistance definitions for systems where multiple coalitions can form. The extension that we provide here is inspired by threshold cryptography, and allows coalitions up to certain thresholds.

Others ╲ This	Diligent	Lazy
All diligent	$r - \mathsf{cost}(1)$	$rq - \big(f + b(n-1)\big)(1-q) - \mathsf{cost}(q)$
k lazy	$r + b(1-q) - \mathsf{cost}(1)$	$rq - \big(f + \frac{b(n-k-1)}{k+1}\big)(1-q) - \mathsf{cost}(q)$
All lazy	$r + b(1-q) - \mathsf{cost}(1)$	$r - \mathsf{cost}(q)$

Table (3) The expected utility of each contractor from choosing diligent or lazy with respect to the other players' chosen strategies (where $0 < k < n$) in the outsourced computation mechanism of [4].

Definition 10 (m-Stable Mechanism)**.** *A joint strategy s_N is an m-stable equilibrium, if for all natural numbers $p \leq |N|$ and for all coalitions $C_1, ..., C_p$ satisfying following conditions:*

- $1 \leq |C_1|, \ldots, |C_p| \leq m$,
- $C_1 \cup \ldots \cup C_p = N$,
- $\forall i, j \in (1, \ldots, p)$ *s.t.* $i \neq j$, $C_i \cap C_j = \emptyset$;

we have that for each $i = 1, ..., p$, *the strategy* s_{C_i} *is weakly dominant for the coalition* C_i *and that* $s_{C_1} \cup \ldots \cup s_{C_p} = s_N$. *Given some desired functionality* $\mathcal{F}$, *we say that* (Π, s_N) *is an* m*-stable mechanism for* $\mathcal{F}$, *if* s_N *is an* m*-stable equilibrium of* Π *and the outcome of* (Π, s_N) *satisfies* $\mathcal{F}$.

We stress that m-stability implies m-repellence. This follows from that in an m-stable mechanism, for any coalition of size up to m the honest strategy is the weakly dominant strategy. As a consequence, if a coalition in this size range deviates from the honest strategy, it ends up with a utility that is less or equal.

In some sense, one may consider ℓ-repellence as similar to Nash equilibrium, and m-stability as similar to weakly dominant strategy equilibrium. ℓ-repellence disincentivizes a rational coalition, only in case the coalition believes that all other parties will play the honest strategy. m-stability, on the other hand, further disincentivizes a coalition from deviating, in case the coalition knows some other deviating coalitions. Even if the coalition anticipates that other players may make honest mistakes (e.g., due to miscalculation) in choosing or playing their strategies, the coalition is still incentivized for honesty. This is why ℓ-repellence does not imply ℓ-stability.

Remark 1. *m-stability necessitates each strategy s_i to be the weakly dominant strategy for C_i, i.e., no matter what all other coalitions or players choose, C_i is always better by choosing s_i. Therefore, unlike extension of ℓ-repellence to (ℓ, t)-resistance, an extended definition for involving Byzantine parties in m-stability is redundant. Recall that t-immunity prevents any harm from arbitrarily deviating parties to players of the honest strategy, hence it may still be separately used.*

5.3 Comparison to Previous Definitions

Although compared to k-resiliency definition (i.e., Definition 5) ℓ-repellence is weaker, it is sufficient in cases where the players are rational in choosing to be part of a coalition. We consider that this is a reasonable assumption in many systems where mechanism design ideas are applied, as the main target of these systems is to enforce the rational players towards the system goal.

As long as ℓ-repellence definition is satisfied in a mechanism, no matter the utility distribution among the coalition players is, either all of them obtain the same utility from a deviant strategy as the one from desired strategy or at least one player obtains less utility. Regarding the former case, k-resiliency also does not provide any protection as well. Regarding the latter case, ℓ-repellence unavoidably entails that particular player not to obey the coalition strategy, which provides enough protection for the mechanism's desired functionality.

We emphasize that k-resiliency and (k, t)-robustness imply k-repellence and (k, t)-resistance, respectively. The proofs of these statements are intuitive, hence we will not provide them here. However, the converses of these statements are not true.

We acknowledge that we are unable to detect trivial implications between t-immunity and our definitions, hence one may need to prove a mechanism separately for them. The only trivial implication is for zero-sum mechanisms given as follows:

Theorem 1. *If (Π, s_N) is a zero-sum t-immune mechanism, then it is also a t-repellent mechanism.*

Proof. Assume that (Π, s_N) is a zero-sum t-immune mechanism. Also, assume that (Π, s_N) is not a t-repellent mechanism. Then, there exists a player set T with some strategy profile s'_T s.t. $|T| \leq t$ and $U_T(s'_T \cup s_{-T}) > U_T(s_N)$. Due to zero-sum property, this results in less total utility for players that are not in T. Therefore, there exists a player i that is not in T, for which $U_i(s'_T \cup s_{-T}) < U_i(s_N)$. This means (Π, s_N) is not a zero-sum t-immune mechanism, which contradicts the assumption in the beginning of this proof. □

Regarding m-stability, we have the following implications for (ℓ, t)-resistance. Note that as above mentioned, m-stability trivially implies m-repellence.

Lemma 1. *If (Π, s_N) is an m-stable mechanism, then it is a $(m, |N| - m)$-resistant mechanism.*

Proof. Assume that (Π, s_N) is an m-stable mechanism. Also, assume that it is not an $(m, |N| - m)$-resistant mechanism. Then, there exist sets $C, T \subseteq N$ such that $C \cap T = \emptyset$, $|C| \leq m$, $|T| \leq |N| - m$, $U_C(s'_C \cup s'_T \cup s_{-(C \cup T)}) > U_C(s'_T \cup s_{-T})$. Therefore, $s_C \subseteq s_N$ is not a weakly dominant strategy for C. This means (Π, s_N) is not an m-stable mechanism, which contradicts the assumption in the beginning of this proof. □

Theorem 2. *If (Π, s_N) is an m-stable mechanism, then $\forall \ell \leq m$ it is an $(\ell, |N| - \ell)$-resistant mechanism.*

Proof. Assume that (Π, s_N) is an m-stable mechanism. Also, assume that $\exists \ell \leq m$ s.t. it is not an $(\ell, |N| - \ell)$-resistant mechanism. Then by Lemma 1, (Π, s_N) is not an ℓ-stable mechanism. As $\ell \leq m$, this means (Π, s_N) is not an m-stable mechanism, which contradicts the assumption in the beginning of this proof. □

6 Example Mechanisms from Literature

In this section, in order to show the usefulness of our definitions, we apply them to some example problems from cloud computing and network security literature. For comparison and reference, we also analyze them with the previous k-resiliency definition.

6.1 Example 1: Incentivized Outsourced Computation

We consider the setting provided in [4]. We briefly describe it as follows. The setting involves a boss and, as players, n rational contractors. The boss has a costly algorithm, of which execution she wants to outsource to the n contractors. Each contractor can choose either the diligent strategy or the lazy strategy. The former means that it runs the correct algorithm whose cost is denoted as $\mathsf{cost}(1)$. The latter means that it runs a less costly deterministic algorithm (called "q algorithm") which gives the correct output with probability q and has a cost denoted as $\mathsf{cost}(q)$. Further, according to [4], the q algorithm run by all lazy players is assumed to be the same. If all contractors return the same output, the boss just accepts it as the correct one and gives the reward r to each of them. Otherwise, the diligent contractors execute a protocol with the boss to catch the lazy ones. In this case, the diligent ones receive the reward r and a bounty b. A share for the total bounties of the diligent contractors plus a fixed fine f is collected from

each contractor. Table 3 provides the resulting expected utility matrix. We note that for this mechanism to be meaningful, it is necessary that $\mathsf{cost}(q) < \mathsf{cost}(1) < r$. [4] has shown that the boss should set $b > r/(1-q)$ to have all diligent as the unique Nash equilibrium (without any restriction on the fine f) if no coalition is allowed. While [4] suggests setting $b \approx r$ is sufficient for practical purposes, our findings in Theorem 4 show that it can even be considered close to optimal, as $b \leq r(n-1)/(n-2)$ guarantees security against large coalitions. Therefore, applying our definitions lead to better understanding of the existing mechanisms not only from a theoretical viewpoint, but also with practical importance in setting the system parameters.

This mechanism is a good example for the limitation of the previous k-resiliency definition, and how it can mark as insecure a mechanism secure against large coalitions (i.e., up to one less than all participants) by our definitions.

Theorem 3. *The incentivized outsourced computation mechanism of [4] is not 2-resilient.*

Proof. Let C be a coalition of two arbitrary players i and j. Let s'_C be the strategy where i and j play the lazy and the diligent strategies, respectively. The utility of j becomes $U_j(s'_C \cup s_{-C}) = r + b(1-q) - \mathsf{cost}(1)$, which is greater than $U_j(s_N) = r - \mathsf{cost}(1)$. □

Regarding our definitions, we only show that $(n-1)$-stability of this mechanism, which implies $(n-1)$-repellence. Further, by Theorem 2, for all $\ell \leq n-1$ it implies$(\ell, n-\ell)$-resistance. The theorem given below concerns only mechanisms with size $n > 2$, since for $n = 2$ and $b > r/(1-q)$, this outsourced computation mechanism is already 1-stable without any upper bound for b (i.e., for each player, the diligent strategy has been shown as weakly dominant by [4]).

Theorem 4. *For $n > 2$, if the boss sets the reward and bounty as $r(n-1)/(n-2) \geq b > r/(1-q)$, the incentivized outsourced computation mechanism of [4] is $(n-1)$-stable.*

Proof. Let C be a coalition of size $|C| \leq n-1$. W.l.o.g, we need to show the all diligent strategy s_C is its weakly dominant strategy.

Case 1. Assume all the players outside C play diligent strategy. Let s_{-C} denote their strategy profile. Also, let $s_N = s_C \cup s_{-C}$. Then, $U_C(s_N) = |C| \cdot \big(r - \mathsf{cost}(1)\big)$. Let s'_C be a strategy profile for C s.t. k players in C play lazy. If the q algorithm returns the correct output (i.e., with probability q), the coalition utility compared to honest strategy only changes due to the decrease in the cost of the executed algorithm by the lazy players. More concretely, the utility of the coalition becomes $U_C(s'_C \cup s_{-C}) = U_C(s_N) + k \cdot \big(r - \mathsf{cost}(q)\big) - k \cdot \big(r - \mathsf{cost}(1)\big) = U_C(s_N) + k \cdot \big(\mathsf{cost}(1) - \mathsf{cost}(q)\big)$. If the q algorithm returns some incorrect output (i.e., with probability $1-q$), the coalition utility compared to honest strategy changes due to failed rewards and paid fines by the lazy players, bounties paid to players outside of the coalition, and the decrease in the cost of the executed algorithm. More concretely, the utility of the coalition becomes $U_C(s'_C \cup s_{-C}) = U_C(s_N) - kr - kf - b(n - |C|) + k \cdot \big(\mathsf{cost}(1) - \mathsf{cost}(q)\big)$. We deduce the expected utility of the coalition as $U_C(s'_C \cup s_{-C}) = U_C(s_N) + k \cdot \big(\mathsf{cost}(1) - \mathsf{cost}(q)\big) + (1-q)\big(-kr - kf - b(n - |C|)\big)$. Then, we calculate $k \cdot \big(\mathsf{cost}(1) - \mathsf{cost}(q)\big) + (1-q)\big(-kr - kf - b(n-|C|)\big) \leq kr + (1-q)\big(-kr - kf - b(n-|C|)\big) \leq kr + (1-q)\big(-kr - b(n-|C|)\big) \leq kb(1-q) + (1-q)\big(-kr - b(n-|C|)\big) = -(1-q)kr + b(1-q)(k-n+|C|)$. For $r \geq \frac{b(k-n+|C|)}{k}$, we obtain $U_C(s'_C \cup s_{-C}) \leq U_C(s_N)$. As these inequalities need to hold for all $k \leq |C|$ and all $|C| < n$, we need to find the maximum of the lower bound of r. For this purpose, we first set $|C| = n-1$, which can be done, since for any possible value of k, the value of $|C|$ can be $n-1$. Then we obtain $r \geq \frac{b(k-1)}{k}$. Again, for

the maximum of the lower bound of r, we set $k = n-1$, which can be done, since we have set $|C| = n-1$ and still have $k \leq |C|$. At the end, we obtain the lower bound of r as $r \geq \frac{b(n-2)}{n-1}$, or alternatively the upper bound of b as $b \leq \frac{r(n-1)}{n-2}$ for $n > 2$.

Case 2. Assume $y > 0$ and $z > 0$ players outside C play lazy and diligent strategies, respectively, s.t. we have $y + z + |C| = n$. Let s'_{-C} denote their strategy profile. Let s'_C be a strategy profile for C s.t. k players in C play lazy. If the q algorithm returns the correct output (i.e., with probability q), the utility of the coalition change due to the decrease in the cost of the executed algorithm by the lazy players in the coalition. More concretely, the utility of the coalition becomes $U_C(s'_C \cup s'_{-C}) = U_C(s_C \cup s'_{-C}) + k \cdot \big(\mathsf{cost}(1) - \mathsf{cost}(q)\big)$. Note that no bounty is received by any player, as the lazy outside the coalition also benefit from the correct output of the q algorithm (as all the players assumed to be running the same deterministic q algorithm). If the q algorithm returns some incorrect output (i.e., with probability $1-q$), the coalition utility changes due to receiving less bounties and rewards, fines paid by the lazy players, bounties paid to the players outside of the coalition, and the decrease in the cost of the executed algorithm by the lazy players. More concretely, the utility of the coalition becomes $U_C(s'_C \cup s'_{-C}) = U_C(s_C \cup s'_{-C}) - kb - kr - kf - bz \cdot \frac{k}{k+y} + k \cdot \big(\mathsf{cost}(1) - \mathsf{cost}(q)\big)$. We deduce the expected utility of the coalition as $U_C(s'_C \cup s'_{-C}) = U_C(s_C \cup s'_{-C}) - (1-q)\Big(kb + kr + kf + bz \cdot \frac{k}{k+y}\Big) + k \cdot \big(\mathsf{cost}(1) - \mathsf{cost}(q)\big)$. Then, we calculate $-(1-q)\Big(kb + kr + kf + bz \cdot \frac{k}{k+y}\Big) + k \cdot \big(\mathsf{cost}(1) - \mathsf{cost}(q)\big) \leq -(1-q)\Big(kb + kr + kf + bz \cdot \frac{k}{k+y}\Big) + kr \leq -(1-q)\Big(kb + kr + kf + bz \cdot \frac{k}{k+y}\Big) + kb(1-q) = -(1-q)\Big(kr + kf + bz \cdot \frac{k}{k+y}\Big) \leq 0$. For all k and $|C|$ in the relevant range, we obtain $U_C(s'_C \cup s'_{-C}) \leq U_C(s_C \cup s'_{-C})$. We stress that the main difference from Case 1 occurs due to loss of the bounties by lazy when playing s'_{-C} in this case.

Case 3. Assume all the players outside C play lazy strategy. Let s''_{-C} denote their strategy profile. We check the deviation strategies in two separate subcases below.

Let s'_C be a strategy profile for C s.t. $k < |C|$ players in C play lazy. If the q algorithm returns the correct output (i.e., with probability q), the coalition utility changes due to the decrease in the cost of the executed algorithm by the lazy players in the coalition. More concretely, the utility of the coalition becomes $U_C(s'_C \cup s''_{-C}) = U_C(s_C \cup s''_{-C}) + k \cdot \big(\mathsf{cost}(1) - \mathsf{cost}(q)\big)$. Note that no bounty is received by any player, as the lazy outside the coalition also benefit from the correct output of the q algorithm (as all the players assumed to be running the same deterministic q algorithm). If the q algorithm returns some incorrect output (i.e., with probability $1-q$), the coalition utility changes due to receiving less bounties and rewards, fines and bounties paid by the lazy players, and the decrease in the cost of the executed algorithm by the lazy players. More concretely, the utility of the coalition becomes $U_C(s'_C \cup s''_{-C}) = U_C(s_C \cup s''_{-C}) - kb - kr - kf - \frac{(|C|-k)bk}{|N-C|+k} + k \cdot \big(\mathsf{cost}(1) - \mathsf{cost}(q)\big)$. Hence, the expected utility of the coalition is $U_C(s'_C \cup s''_{-C}) = U_C(s_C \cup s''_{-C}) - (1-q)\Big(kb + kr + kf + \frac{(|C|-k)bk}{|N-C|+k}\Big) + k \cdot \big(\mathsf{cost}(1) - \mathsf{cost}(q)\big)$. We calculate $-(1-q)\Big(kb + kr + kf + \frac{(|C|-k)bk}{|N-C|+k}\Big) + k \cdot \big(\mathsf{cost}(1) - \mathsf{cost}(q)\big) \leq -(1-q)(kb + kr + kf) + kr \leq -(1-q)(kb + kr + kf) + kb(1-q) = -(1-q)(kr + kf) \leq 0$. Therefore, for all k and $|C|$ in the relevant range, we obtain $U_C(s'_C \cup s''_{-C}) \leq U_C(s_C \cup s''_{-C})$.

Now, let s'_C be a strategy profile for C s.t. all players in C play lazy. We use the coalition utility $U_C(s_N)$ from all-diligent as reference to obtain $U_C(s'_C \cup s''_{-C}) = U_C(s_N) + |C| \cdot \big(\mathsf{cost}(1) - \mathsf{cost}(q)\big) = U_C(s_C \cup s''_{-C}) - |C|b(1-q) + |C| \cdot \big(\mathsf{cost}(1) - \mathsf{cost}(q)\big)$. As $\mathsf{cost}(1) - \mathsf{cost}(q) \leq r \leq b(1-q)$,

for all $|C|$ in the relevant range, we have $U_C(s'_C \cup s''_{-C}) \leq U_C(s_C \cup s''_{-C})$. □

Theorem 5. *The incentivized outsourced computation mechanism of [4] is* $(n-1)$*-immune.*

Proof. Since there exists at least one player acting honestly, the all lazy utility cannot be achieved. If all the players act diligently, the utility of an honest player is $r - \mathsf{cost}(1)$. For any other outcome, the utility of an honest player turns out $r + b(1-q) - \mathsf{cost}(1)$, which is greater than the former. □

Remark 2. *The incentivized outsourced computation mechanism of [4] analyzed above cannot trivially be altered to satisfy k-resiliency by replacing the all diligent strategy utility (i.e.,* $r - \mathsf{cost}(1)$*) with* $(r + b(1-q) - \mathsf{cost}(1))$*, since this means that the boss hands* $r + b(1-q)$ *as reward to each contractor in case of consensus of the output. In turn, this would enforce the all lazy utility to become* $r + b(1-q) - \mathsf{cost}(q)$*, providing another Nash equilibrium to the system, i.e., all lazy. Yet, this mechanism still satisfies our proposed definitions, and provides strong security against coalitions with up to* $n-1$ *rational players.*

6.2 Example 2: Forwarding Dilemma

We consider the "forwarding dilemma" introduced by [12] to model forwarding of a flooded packet in a wireless ad hoc network. We briefly describe it as follows. In this game, players are network nodes who has received the same flooded packet. Regarding this packet, each player has two strategies: forward it or drop it. [12] shows that desirable strategies of this game are those with one player that plays forward, while the rest plays drop. Here, the number of forwarding players needs limiting to 1 in order to avoid excess bandwidth overhead.

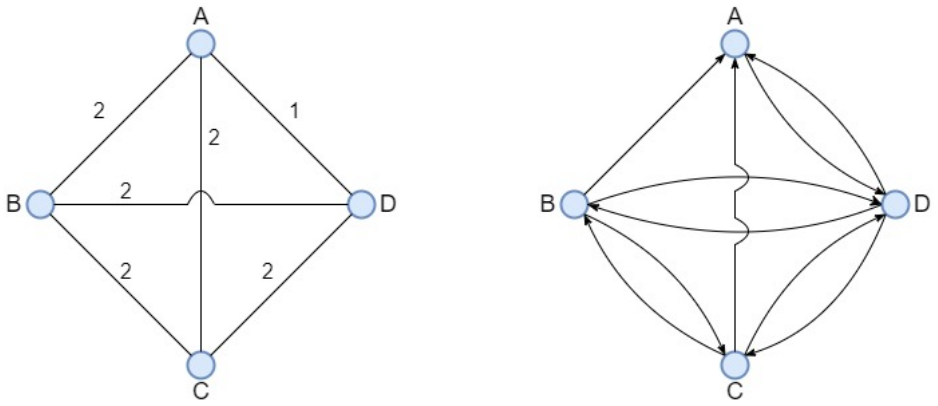
There are two values that affect the utility of each player, namely the network gain factor g and the forwarding cost c. The utilities of the players are defined according to her and other players' strategies as in Table 4. Obviously, $c < g$ is required for incentivizing one forward and the rest drop.

Others ╲ This	forward	drop
All drop	$g - c$	0
At least one forward	$g - c$	g

Table (4) The utility of each player from choosing forward or drop with respect to the other players' chosen strategies in the forwarding dilemma of [12].

In the beginning of each game a mediator assigns forwarding to one of the player and dropping to the remaining players [1]. According to [12], this strategy profile is a Nash equilibrium. Let N be the set of players with $|N| = n$ and P be the player who is assigned forwarding by the mediator.

[1] In mechanism design, a mediator is generally used for coordination of players in playing an equilibrium, in case there exist multiple equilibria. The mediator's job is just to assign a strategy profile for all players in the beginning of a game. We highlight that it is not an authority and that it cannot enforce any particular strategy to any player. A mediator can even be implemented as a deterministic algorithm that is executed by each player on some common input before the game starts. Regarding the forwarding dilemma, the algorithm may select one of the players for forward strategy (e.g., a linear search for the player with the smallest address or id encoding). Clearly, choice of the mediator and its existence does not affect our analysis and results.



(a) Initial graph H (b) Induced graph G

Figure (1) Graphs related to an instance of the strong connectivity game of [11]. The one given in (a) defines the game, and the one given in (b) shows the induced graph if the players A, B, C, and D choose radii 1, 2, 2, and 2, respectively.

We show that according to k-resiliency definition, this mechanism is not incentive compatible even in the presence of coalitions with 2 players.

Theorem 6. *The given mechanism of forwarding dilemma game is not 2-resilient.*

Proof. Let $C \subseteq N$ consist of P and another player, say P'. If P drops and P' forwards, the utility of P increases to g from $g - c$. Thus, this mechanism is not 2-resilient. □

On the other hand, this mechanism is secure in the presence of a single coalition, according to our ℓ-repellent definition.

Theorem 7. *The given mechanism of forwarding dilemma game is n-repellent.*

Proof. We show that any coalition cannot receive more utility than the honest strategy. W.l.o.g, let $C \subseteq N$ be a coalition of size at most n. Assume players outside of the coalition follow mediator.

Case 1. Assume P is not in C. Then since each player in the C already receives g, the payoff of the coalition cannot increase by any change of the strategy of the coalition.

Case 2. Assume P is in C. If they follow the honest strategy, the payoff of the coalition is $|C| \cdot g - c$. If nobody in the coalition forwards, the payoff of the coalition is 0. If more than one player in the coalition forwards, the payoff of the coalition is at most $|C| \cdot g - 2 \cdot c$. Thus, there is no strategy for the coalition C such that the coalition receives greater payoff than the honest strategy. □

We again observe a good example mechanism that is marked completely insecure against a coalition (not even 2-resilient) by the definition of [7]. However, by our ℓ-repellence definition, it is secure no matter what the size of the coalition is. For completeness, we also show that unfortunately this game does not satisfy our other definitions and t-immunity.

Theorem 8. *The given mechanism of forwarding dilemma game is not* $(1, 1)$*-resistant.*

Proof. Let C be any coalition of size 1 and $P \notin C$. Also, let $T = (P)$. If P deviates and chooses to drop, then C receives 0 by following the mediator while its utility becomes $g - c$, when the player in the coalition deviates and forwards. □

Theorem 9. *The given mechanism of forwarding dilemma game is not* 1*-immune.*

Proof. If P deviates by dropping, then the utility of each player from the honest strategy decreases to 0. □

Theorem 10. *The given mechanism of forwarding dilemma game is not* 1*-stable.*

Proof. As neither forwarding nor dropping is a weakly dominant strategy, clearly we do not have 1-stability in this game. □

6.3 Example 3: Strong Connectivity

We consider the "strong connectivity" game investigated by [11] for connectivity problems in an ad hoc network where nodes are selfish. We summarize their results related to our work as follows. They observe wireless networks with omni-directional antennas. The network topology can be represented by an undirected, weighted, complete graph $H(V, E', w)$, where V denotes the vertices with $|V| = n$, E' denotes edges and w is a weight vector with w_e is weight of an edge $e \in E'$.

The players in this game are the nodes of the graph H. Strategy of each player is choosing a radius r. Radii choices of the nodes induce a directed, unweighted graph $G(V, E)$ such that $e = (u, v) \in E$ if and only if $r_u \geq w_e$. We say that a node u can reach another node v, if there exists a path between them in the induced graph. Then, the utility of a node v is defined as $-r_v^\alpha$ if v can reach every other node for some constant distance power gradient α. If there is a node that v cannot reach, its utility is $-\infty$.

The main goal of the network designer here is strong connectivity, i.e. each node can reach any other node. Here, we consider an instance of strong connectivity game with four nodes $\mathsf{A}, \mathsf{B}, \mathsf{C}$ and D and the initial graph shown in Figure 1a. There exists a mediator who assigns a radius to each player [2]. Specifically, here it assigns radii $(1, 2, 2, 2)$ to $\mathsf{A}, \mathsf{B}, \mathsf{C}, \mathsf{D}$. The induced graph G for this game is given in Figure 1b.

Theorem 11. *The given mechanism of strong connectivity game is not 2-resilient.*

Proof. Consider the coalition $C = (\mathsf{A}, \mathsf{D})$. If this mechanism were 2-resilient, A choosing 1 and D choosing 2 as radius would be a group best response. For this strategy, utility of D is -2^α. On the other hand, if D chooses 1 and A chooses 2 as radius, utility of D becomes -1^α, which is greater than -2^α. Hence, the former group strategy is not a group best response. Consequently, this mechanism is not 2-resilient. □

Theorem 12. *The given mechanism of strong connectivity game is* 4*-repellent.*

Proof. Assume that this mechanism is not 4-repellent. Then, there exists a coalition C with a size up to 4 s.t. there exists a strategy that makes the utility of C higher. We note that in this mechanism, any node other than D cannot decrease her utility without losing its all

[2] In line with the mediator of forwarding dilemma, here it can again be a trivial deterministic algorithm executed by each player on some common input in the beginning.

connections. Thus, D must be a part of the coalition, and must decrease her radius to 1. Note that D cannot choose a radius lower than 1, since it would prevent her from connecting to any other node. When D decreases her radius to 1, she can only connect to A. Thus, for them to reach other nodes A must be a part of the coalition and it should change her strategy as well. The only reasonable joint deviation is A choosing 2 and D choosing 1. However, for this deviation the utility of the coalition does not change. By contradiction we have that this mechanism is 4-repellent. □

Observe that this mechanism is not 2-resilient, yet it is repellent against even a coalition of all the players. For completeness, we also analyze it with our other definitions and t-immunity.

Theorem 13. *The given mechanism of strong connectivity game is not* $(1,1)$*-resistant.*

Proof. Let D be a Byzantine player and choose 1 as radius. Then, A cannot connect to B or C by choosing radius 1, which results with a payoff $-\infty$ for A. On the other hand, if A chooses 2 as radius, her payoff is -2^{α} which is greater than $-\infty$. As a coalition can consist of only A, this mechanism is not $(1,1)$-resistant. □

Theorem 14. *The given mechanism of strong connectivity game is not* 1*-immune.*

Proof. Let D be an arbitrarily acting player. Let D choose 1 as radius. Then, A cannot connect to B or C by choosing radius 1, which results with a payoff $-\infty$ for A. Thus, this mechanism is not t-immune for any $t \geq 1$. □

Theorem 15. *The given mechanism of strong connectivity game is not* 1*-stable.*

Proof. For each of the players A and D, there exists no weakly dominant strategy. □

7 Conclusion

This paper paves the way toward incorporating multiple coalitions into game-theoretic definitions to capture realistic scenarios in cloud, computer, and network security. Our definitions are inspired by those of threshold cryptography, and hence enables security against rational coalitions of m players each. We also show, in the paper, that previous definitions bridging game theory and cryptography, specifically k-resiliency and (k,t)-robustness, are not immediately useful, or directly applicable to some applications in many scenarios. We then combine transferable utility with threshold cryptography to achieve the desired novel definitions. Finally, we show the applicability and usefulness of our novel definitions in three different games in the area of cloud and network security. We hope that our definitions will help researchers design their mechanisms practically, and show their security with ease and with easily understandable goals.

Designs and analyses of novel mechanisms from more diverse areas by using our definitions seem as interesting research. These areas include (but not limited to) secure outsourced computation [51], secure multi-party computation (as in [7] or combination of game theory with known secure protocols [52] for efficiency), more dependable and multi-party versions of private function evaluation [53, 54, 55], ad hoc network security [11, 12], Byzantine fault tolerant systems [37, 56] where rational and collaborating processors are involved, blockchain mining (as in [57]), and mining pool games (as in [58, 59]). We note that further definitions incorporating ours for more specific or sophisticated use cases, such as modelling conflict of interests, are left as future work as well.

References

[1] M. H. Manshaei, Q. Zhu, T. Alpcan, T. Başar, and J.-P. Hubaux, "Game theory meets network security and privacy," *ACM Comput. Surv.*, July 2013.

[2] J. Katz, "Bridging game theory and cryptography: Recent results and future directions," in *TCC '08*, 2008.

[3] M. Belenkiy, M. Chase, C. C. Erway, J. Jannotti, A. Küpçü, and A. Lysyanskaya, "Incentivizing outsourced computation," in *NetEcon '08*, 2008.

[4] A. Küpçü, "Incentivized outsourced computation resistant to malicious contractors," *IEEE TDSC*, vol. 14, no. 6, pp. 633–649, 2017.

[5] S. Avizheh, M. Nabi, R. Safavi-Naini, and M. Venkateswarlu K., "Verifiable computation using smart contracts," in *ACM CCSW '19*, 2019.

[6] A. Küpçü and R. Safavi-Naini, "Smart contracts for incentivized outsourcing of computation," in *ESORICS CBT '21*, 2021.

[7] I. Abraham, D. Dolev, R. Gonen, and J. Halpern, "Distributed computing meets game theory: Robust mechanisms for rational secret sharing and multiparty computation," in *ACM PODC '06*, 2006.

[8] I. Abraham, D. Dolev, and J. Y. Halpern, "Distributed protocols for leader election: A game-theoretic perspective," in *DISC '12*, 2013.

[9] S. Bag, S. Ruj, and K. Sakurai, "Bitcoin block withholding attack: Analysis and mitigation," *IEEE TIFS*, vol. 12, pp. 1967–1978, Aug 2017.

[10] A. Kothapalli, A. Miller, and N. Borisov, "Smartcast: An incentive compatible consensus protocol using smart contracts," in *FC '17*, 2017.

[11] S. Eidenbenz, V. S. A. Kumar, and S. Zust, "Equilibria in topology control games for ad hoc networks," in *DIALM-POMC '03*, 2003.

[12] M. Naserian and K. Tepe, "Game theoretic approach in routing protocol for wireless ad hoc networks," *Ad Hoc Networks*, vol. 7, pp. 569–578, 2009.

[13] Z. Tian, X. Gao, S. Su, J. Qiu, X. Du, and M. Guizani, "Evaluating reputation management schemes of internet of vehicles based on evolutionary game theory," *IEEE Trans. Veh*, vol. 68, no. 6, 2019.

[14] S. Kamara and A. Küpçü, "Dogfish: Decentralized optimistic game-theoretic file sharing," in *ACNS '18*, 2018.

[15] Y. Wang, H. Wang, and Q. Xu, "Rational secret sharing with semi-rational players," *Int. J. Grid Util. Comput.*, vol. 3, p. 59–67, Mar. 2012.

[16] R. Brenguier, "Robust equilibria in mean-payoff games," in *FoSSaCS '16*, 2016.

[17] J. Y. Halpern and X. Vilaça, "Rational consensus: Extended abstract," in *ACM PODC '16*, 2016.

[18] A. De Santis, Y. Desmedt, Y. Frankel, and M. Yung, "How to share a function securely," in *STOC '94*, 1994.

[19] Y. Desmedt, "Threshold cryptography," in *Encyclopedia of Cryptography and Security*, Springer US, 2011.

[20] S. D. Gordon and J. Katz, "Rational secret sharing, revisited," in *SCN '06*, 2006.

[21] A. Lysyanskaya and N. Triandopoulos, "Rationality and adversarial behavior in multi-party computation," in *CRYPTO '06*, 2006.

[22] G. Fuchsbauer, J. Katz, and D. Naccache, "Efficient rational secret sharing in standard communication networks," in *TCC '10*, 2010.

[23] V. Dani, M. Movahedi, Y. Rodriguez, and J. Saia, "Scalable rational secret sharing," in *ACM PODC '11*, 2011.

[24] J. Upadhyay, "Integrity and privacy of large data," 2015.

[25] A. S. Aiyer, L. Alvisi, A. Clement, M. Dahlin, J.-P. Martin, and C. Porth, "Bar fault tolerance for cooperative services," *ACM SIGOPS Oper. Syst. Rev.*, vol. 39, no. 5, p. 45–58, 2005.

[26] H. C. Li, A. Clement, E. L. Wong, J. Napper, I. Roy, L. Alvisi, and M. Dahlin, "Bar gossip," in *OSDI '06*, 2006.

[27] M. Feldman, C. Papadimitriou, J. Chuang, and I. Stoica, "Free-riding and whitewashing in peer-to-peer systems," in *ACM PINS '04*, 2004.

[28] J. Garay, J. Katz, U. Maurer, B. Tackmann, and V. Zikas, "Rational protocol design: Cryptography against incentive-driven adversaries," in *IEEE FOCS '13*, 2013.

[29] Y. Heller, "Ex-ante and ex-post strong correlated equilbrium," tech. rep., University Library of Munich, Germany, 2008.

[30] J. Farrell and M. Rabin, "Cheap talk," *J. Econ. Perspect.*, vol. 10, no. 3, 1996.

[31] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. `http://bitcoin.org/bitcoin.pdf`.

[32] V. Buterin, "Ethereum white paper: A next-generation smart contract and decentralized application platform," 2013.

[33] `https://litecoin.org/`, 2011. Accessed: 2021-03-15.

[34] C. Dwork and M. Naor, "Pricing via processing or combatting junk mail," in *CRYPTO '92*, 1993.

[35] S. King and S. Nadal, "Pcoin: Peer-to-peer crypto-currency with proof-of-stake," in *Peercoin Whitepaper*, 2012.

[36] Y. Hassanzadeh-Nazarabadi, A. Küpçü, and Ö. Özkasap, "Lightchain: Scalable dht-based blockchain," *IEEE TPDS*, 2021.

[37] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," *ACM Trans. Program. Lang. Syst.*, vol. 4, July 1982.

[38] M. Lokhava, G. Losa, D. Mazières, G. Hoare, N. Barry, E. Gafni, J. Jove, R. Malinowsky, and J. McCaleb, "Fast and secure global payments with stellar," in *ACM SOSP '19*, 2019.

[39] S. Popov, "The tangle," 2017. http://www.iota.org/IOTA_Whitepaper.pdf.

[40] Y. Sompolinsky, S. Wyborski, and A. Zohar, "Phantom and ghostdag: A scalable generalization of nakamoto consensus," 2018. https://eprint.iacr.org/2018/104.pdf.

[41] Y. Sompolinsky, Y. Lewenberg, and A. Zohar, "Spectre : Serialization of proof-of-work events : Confirming transactions via recursive elections," 2016. https://eprint.iacr.org/2016/1159.pdf.

[42] M. Y. Afanasev, Y. V. Fedosov, A. A. Krylova, and S. A. Shorokhov, "An application of blockchain and smart contracts for machine-to-machine communications in cyber-physical production systems," in *IEEE ICPS '18*, 2018.

[43] N. Atzei, M. Bartoletti, and T. Cimoli, "A survey of attacks on ethereum smart contracts sok," in *POST '17*, 2017.

[44] N. Atzei, M. Bartoletti, T. Cimoli, S. Lande, and R. Zunino, "Sok: Unraveling bitcoin smart contracts," in *POST '18*, 2018.

[45] Z. Zheng, S. Xie, H.-N. Dai, W. Chen, X. Chen, J. Weng, and M. Imran, "An overview on smart contracts: Challenges, advances and platforms," *Future Generation Computer Systems*, vol. 105, pp. 475 – 491, 2020.

[46] H. Peters, "Cooperative games with transferable utility," in *Game Theory: A Multi-Leveled Approach*, Springer Berlin Heidelberg, 2008.

[47] J. R. Douceur, "The sybil attack," in *IPTPS '02*, 2002.

[48] R. J. Aumann, "16. acceptable points in general cooperative n-person games," in *Contributions to the Theory of Games (AM-40), Volume IV*, Princeton University Press, 1959.

[49] B. D. Bernheim, B. Peleg, and M. Whinston, "Coalition-proof nash equilibria i. concepts," *Journal of Economic Theory*, vol. 42, no. 1, pp. 1–12, 1987.

[50] L. Shapley and M. Shubik, "On market games," *Journal of Economic Theory*, vol. 1, no. 1, pp. 9–25, 1969.

[51] Y. Qian, Y. Zhang, X. Chen, and C. Papamanthou, "Streaming authenticated data structures: Abstraction and implementation," in *ACM CCSW '14*, 2014.

[52] N. P. Smart and T. Tanguy, "Taas: Commodity mpc via triples-as-a-service," in *ACM CCSW '19*, 2019.

[53] M. A. Bingöl, O. Biçer, M. S. Kiraz, and A. Levi, "An Efficient 2-Party Private Function Evaluation Protocol Based on Half Gates," *The Computer Journal*, 2018.

[54] S. Felsen, A. Kiss, T. Schneider, and C. Weinert, "Secure and private function evaluation with intel sgx," in *ACM CCSW '19*, 2019.

[55] O. Biçer, M. A. Bingöl, M. S. Kiraz, and A. Levi, "Highly efficient and re-executable private function evaluation with linear complexity," *IEEE TDSC*, 2020.

[56] C. Cachin and B. Tackmann, "Asymmetric distributed trust," in *OPODIS '19*, 2019.

[57] I. Tsabary and I. Eyal, "The gap game," in *ACM CCS '18*, 2018.

[58] Z. Chen, X. Sun, X. Shan, and J. Zhang, "Decentralized mining pool games in blockchain," in *IEEE ICKG '20*, 2020.

[59] W. Li, M. Cao, Y. Wang, C. Tang, and F. Lin, "Mining pool game model and nash equilibrium analysis for pow-based blockchain networks," *IEEE Access*, 2020.